A Distributed Collaborative Relay Protocol for Multi-hop WLAN Accesses

Bo Zhang*, Zhongming Zheng*, Xiaohua Jia†, Kan Yang*
Department of Computer Science
City University of Hong Kong
*{Bo.Zhang, 50541654, kanyang3}@student.cityu.edu.hk
†jia@cs.cityu.edu.hk

Abstract—Due to the packet-fairness property of the 802.11 WLANs, the clients with low rates take longer time to transmit a packet and would reduce the overall network throughput, as well as the per-node throughput of those high data rate clients. We propose a collaborative relay method, which allows high data rate clients to relay traffic for the low data rate clients to improve the network throughput. There are three original contributions made by this paper: 1) we develop a generalized model to analyze multi-hop relay for clients. By using this model, clients that are several hops away from each other may take the advantage of concurrent transmissions. 2) We propose a centralized algorithm to compute multi-hop relay topology for the clients and it can achieve the optimal throughput in our defined model. 3) We propose a distributed multi-hop relay protocol for clients dynamically connected to or disconnected from the network. Simulation results show that our algorithm can improve the throughput up to 42% on average, which is significantly better than the previous work. Up to 75% of the clients in the network use 1-hop or multi-hop relay, instead of connecting to the AP directly, for the performance gain; and up to 39% of the links have the opportunities to transmit concurrently with another link.

I. INTRODUCTION

The data rate of a client to access AP is determined by the signal quality between the client and the AP. The clients having poor signal strength from AP (they are usually further away from the AP) have low data rate. The built-in auto-rate fallback technology in IEEE 802.11 can detect the receiving signal strength and automatically increase or decrease data rate accordingly. However, the 802.11 network is designed based on packet-fairness among clients. That is, all clients have equal chance to access the wireless channel, and once a client seizes the channel, it keeps occupying the channel until it finishes the transmission of the whole packet. In most 802.11 protocols, the highest data rate can be about 10 times higher than the lowest. A low data rate client takes up a much larger fraction of the total radio airtime than the clients with high data rate, and drag down the overall system throughput significantly. This is called rate anomaly problem in 802.11 protocols. Considering an example that 3 clients are connected to an AP at the data rate of 54Mbps, if there is no packet overhead, the throughput of each client is 18Mbps theoretically. Once a client moves further away and can only communicate with AP at 6Mbps. Due to the packet-fairness property, the per-client throughput drops to 4.9Mbps.

One way to solve the rate anomaly problem is to let intermediate nodes relay traffic for the clients that are far away from the AP. Fig.1 is a set of experimental data supplied by Huawei, which shows the relationship between transmission distance and data rate. A node, say $v_A$, that is far away from the AP can only operate at rate 6Mbps. If we let a client $v_B$, which is in the middle between $v_A$ and AP, relay traffic for $v_A$, the data rate of $v_A$ would be 12Mbps, assuming the data rate between $v_A$ and $v_B$ is also 24Mbps. By helping $v_A$ to relay the traffic, $v_B$ in fact can also gain better throughput, compared with the case letting $v_A$ connect to the AP directly.

Extensive studies have been done on the use of intermediate nodes for traffic relay to solve the rate anomaly problem [1]–[3]. Due to the heavy transmission overhead introduced by multi-hop relay, most of these studies focused on only 1-hop relay. That is, a client can at most use one relay node to relay its traffic to AP. However, if we take concurrent transmissions into consideration, by using only 1-hop relay, clients have little chance to transmit concurrently, because the hops are usually long and may interfere each other due to the limit of only 1-hop relay.

In this paper, we extend the 1-hop relay to multi-hop relay. We first propose a general model to analyze the throughput gain of multi-hop relay. Then, we propose an algorithm that can compute the multi-hop relay topology to achieve the optimal system throughput. By allowing multi-hop relay, there are more links (clients) that can transmit concurrently in the system, which further improves the system performance. We also propose a distributed multi-hop relay protocol for clients dynamically connected to or disconnected from the network.

The rest of the paper is organized as follows. In section II, we give an overview of existing work that aims to alleviate rate
anomaly problem. In section III, we describe the architecture overview. A centralized algorithm and the corresponding distributed protocol is proposed in section IV and V respectively, and the simulation results are shown in section VI. Finally, we conclude the paper in section VII.

II. RELATED WORK

Rate anomaly problem was firstly discussed and analyzed in [4] by Heusse et al. Some work suggested changing the MAC to be “time-fair” rather than the current “packet-fair” scheme in [5]–[7] to solve the rate anomaly problem. However, the solutions need to change the MAC layer of the current network, and further degrade the performance of the low-rate clients, which is not a global optimization solution. The idea of letting some clients in WLANs relay traffic for others was presented in [1] and [2], which is called multi-hop extensions. However, they only used 1-hop forwarding, and did not provide any generalized algorithm for the relay network establishment.

Some initial work for multi-hop architecture can be found in [8], [9], and a general multi-hop WLAN architecture was proposed in [10]. However, the rate anomaly problem was not addressed in these paper.

Several practical work for collaborative relay method were implemented in [3], [11]–[13], but none of the above work takes concurrent transmissions into consideration. Since multi-hop relay causes the heavy transmission overheads, most algorithms are designed based on 1-hop relay. Besides that, some algorithms are too complicated to be implemented in real situation.

III. PROBLEM FORMULATION

A. Transmission and Interference Model

As is shown in Fig.2, there are $n$ clients $v_1...v_n$ and one AP in the collaborative relay network (CRN). We assume the transmission power for each client and AP is fixed. Power adjustment is not considered in our paper. Two nodes $v_i$ and $v_j$ can communicate with each other if they are within each other’s transmission range. The data rate $R(v_i, v_j)$ of link $(v_i, v_j)$ is determined by the receiving signal strength, which can be calculated by the transmission power $P_t(v_i)$ and signal attenuation rate $A(v_i, v_j)$ as shown below:

$$R(v_i, v_j) = f(P_t(v_i) \times A(v_i, v_j)), \quad (1)$$

where $f()$ is a mapping from receiving signal strength to data rate. When transmission power is 20dBm, $f()$ can be represented by Table I. The experiment result from Huawei shows the relationship between transmission range and data rate without considering obstacles.

In this paper, we assume communication links are bidirectional, which complies with 802.11 protocols. We also assume the two end nodes of a link use the same data rate, i.e., $R(v_i, v_j) = R(v_j, v_i)$, which basically allows the high rate end node follows the low-rate of the other end node over a link.

We define link interference as follows. Link $l_i$ and $l_j$ interfere with each other if either end node of one link is in the interference range of either end node of the other link.

In our multi-hop relay method, two links that do not interfere with each other can transmit concurrently, which can significantly improve the network throughput.

B. Packet Transmission Time and Network Efficiency

The client-AP packet transmission time is the transmission time of a packet between AP and the end user, which includes the total relay time by all the intermediate relay nodes. Let $T(v_i)$ denote the packet transmission time between AP and client $v_i$, and $t(v_i, v_j)$ the transmission time of a packet over link $(v_i, v_j)$.

$$t(v_i, v_j) = \frac{C}{R(v_i, v_j)} + \tau, \quad (2)$$

where $C$ is the packet size, and $\tau$ is the overhead for each transmission, such as DIFS, SIFS, Frame Header, and ACK frame used by the 802.11 MAC layer. In order to simplify the calculations, we do not take the backoff time for the channel contention period into consideration. In 802.11g BSS mode, $\tau$ is 94$\mu$s, and not influenced by data rate. By using frame bursting transmission technique in 802.11e QoS specification, $\tau$ can be as low as 76$\mu$s.

If a client $v_i$ is connected to the AP directly, the packet transmission time $T(v_i) = t(AP, v_i)$. When $v_i$ is relayed by an intermediate node $v_k$, $T(v_i) = T(v_k) + t(v_k, v_i)$.

Let $\Gamma$ denote the total time for all the clients transmitting one packet in the network. Since we consider multi-hop relay in the system, there are some links that can transmit

<table>
<thead>
<tr>
<th>RX Power (dBm)</th>
<th>Rate (Mbps)</th>
<th>Range (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-68</td>
<td>54</td>
<td>15</td>
</tr>
<tr>
<td>-69</td>
<td>48</td>
<td>16</td>
</tr>
<tr>
<td>-72</td>
<td>36</td>
<td>21</td>
</tr>
<tr>
<td>-77</td>
<td>24</td>
<td>27</td>
</tr>
<tr>
<td>-80</td>
<td>18</td>
<td>33</td>
</tr>
<tr>
<td>-82</td>
<td>12</td>
<td>38</td>
</tr>
<tr>
<td>-84</td>
<td>9</td>
<td>43</td>
</tr>
<tr>
<td>-90</td>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>-92</td>
<td>Interference</td>
<td>73</td>
</tr>
</tbody>
</table>

Fig. 2. A client collaborative relay network.
data concurrently. Let $\delta$ denote the total time saved by the concurrent transmissions during the period that every client sends out a packet. The $\Gamma$ should be:

$$\Gamma = \sum_{i=1}^{n} T(v_i) - \delta. \quad (3)$$

Since 802.11 uses packet-fairness for channel access, the throughput of each client becomes almost the same in a long period of time. Thus, the per-client throughput can be calculated as:

$$P = \frac{C}{\Gamma}. \quad (4)$$

From (3) and (4), we can see that in order to improve the per-client throughput, the total transmission time $\Gamma$ should be decreased, which can be achieved by reducing $T(v_i)$ or increasing $\delta$. Since $\delta$ is largely dependent on the scheduling of transmissions and we do not consider scheduling in this paper, our method will focus on reducing $T(v_i)$ by using multi-hop relays. Nevertheless, concurrent transmissions are allowed in our protocol and is evaluated in the simulations.

IV. CENTRALIZED COLLABORATIVE RELAY ALGORITHM

As discussed in section III, we firstly propose an algorithm to establish a topology which can achieve minimum total transmission time. After that, a simple method is designed for the network to estimate the total time saved by the concurrent transmissions.

Let $T^*(v_i)$ denote the optimal packet transmission time between AP and client $v_i$, we can use the following equation to get a possible route for $v_i$ and compute its transmission time:

$$T'(v_i) = \min\{t(AP, v_i), \min_{1 \leq k \leq n, k \neq i} \{T^*(v_k) + t(v_k, v_i)\}\}. \quad (5)$$

The equation means that the route can be the direct connection between client $v_i$ and AP, or through another optimal link between client $v_k$ and AP.

**Lemma 1.** Equation (5) always produces the optimal $T'(v_i)$ which is equal to $T^*(v_i)$.

**Proof:** Suppose $(AP, v_{k1}, v_{k2}, ..., v_{kn}, v_i)$ is the route that achieve optimal packet transmission time $T^*(v_i)$. It is obvious that $(AP, v_{k1}, v_{k2}, ..., v_{kn})$ must be the route that achieve optimal packet transmission time $T^*(v_{kn})$. Therefore, equation (5) covers all the possibilities of the routes, and can achieve optimal solution.

To establish a collaborative relay network, our task is to construct a tree topology, rooted from AP. Let $T_{AP}$ denote the set of the clients already connected to the tree, and $V$ the rest clients. Suppose for every client $v_i \in V$ and $v_k \in T_{AP}$, it satisfies $T^*(v_i) \geq T^*(v_k)$, then we can use the following equation to get a possible route for any node $v_i$.

$$T(v_i) = \min\{t(AP, v_i), \min_{v_k \in T_{AP}} \{T^*(v_k) + t(v_k, v_i)\}\}. \quad (6)$$

**Lemma 2.** $T(v_x) = T^*(v_x)$, if $T^*(v_x) = \min_{v_i \in V} T^*(v_i)$.

**Proof:** The difference between (5) and (6) is that only part of the $T^*(v_i)$ are considered when calculating $T(v_i)$. Obviously, if $T^*(v_i) = T^*(v_k) + t(v_k, v_i)$, it satisfy $T^*(v_i) \geq T^*(v_k)$, which is just the method of calculating $T(v_x)$ in lemma 2. 

**Lemma 3.** $T^*(v_x) = \min_{v_i \in V} T^*(v_i)$, if $T(v_x) = \min_{v_i \in V} T(v_i)$, $v_x \in V$.

**Proof:** Suppose $T^*(v_x) \neq \min_{v_i \in V} T^*(v_i)$, there must exist a client $v_y \in V$ satisfies $T^*(v_y) = \min_{v_i \in V} T^*(v_i)$. According to lemma 2, $T(v_y) = T^*(v_y)$. The relationship between $T(v_x)$ and $T^*(v_y)$ should be, $T(v_x) < T(v_y) = T^*(v_y) < T^*(v_x)$. Since $T^*(v_x)$ is the optimal packet transmission time, $T(v_x)$ cannot achieve better solution than the optimal value. Therefore, the proposition is wrong. $T^*(v_x)$ must be the minimum among all the clients in set $V$.

According to lemma 2 and lemma 3, an algorithm for calculating $T^*(v_i)$ and constructing the tree topology is listed below.

**Algorithm 1 Algorithm to Construct CRN**

$$T_{AP} \leftarrow \emptyset, V \leftarrow \{v_1, ..., v_n\};$$

while $V \neq \emptyset$ do

- Calculate $T(v_i)$ for all $v_i \in V$;
- Find the client $v_x \in V$ that satisfies $T(v_x) = \min_{v_i \in V} T(v_i)$;
- $T^*(v_x) \leftarrow T(v_x)$;
- Connect $v_x$ to its relay node in $T_{AP}$;
- Add $v_x$ to $T_{AP}$ and delete it from $V$;

end while

**Theorem 4.** Algorithm 1 always produces the optimal $T^*(v_i)$ for any client $v_i$ in the network.

**Proof:** At first, when $T_{AP} = \emptyset$, for the client $v_x$ which satisfies $T(v_x) = \min_{v_i \in V} T(v_i)$, we can get $T^*(v_x) = t(AP, v_x)$. Obviously, $T^*(v_x)$ is optimal for client $v_x$. Our algorithm adds $v_x$ to set $T_{AP}$, and deletes it from set $V$. The supposition $\max_{v_i \in T_{AP}} T^*(v_i) \leq \min_{v_i \in V} T^*(v_i)$ holds.

For the following steps, when there are $m$ clients in set $T_{AP}$, we assume that the supposition $\max_{v_i \in T_{AP}} T^*(v_i) \leq \min_{v_i \in V} T^*(v_i)$ holds. According to lemma 2 and lemma 3, we can get minimal $T^*(v_x)$ by finding the minimal $T(v_x)$ in set $V$. Then, we add client $v_x$ to set $T_{AP}$ and delete it from set $V$. Therefore, when there are $m + 1$ clients in set $T_{AP}$, the supposition $\max_{v_i \in T_{AP}} T^*(v_i) \leq \min_{v_i \in V} T^*(v_i)$ still holds.

In conclusion, the supposition $\max_{v_i \in T_{AP}} T^*(v_i) \leq \min_{v_i \in V} T^*(v_i)$ always holds in Algorithm 1, which means we can always get optimal $T^*(v_i)$ in Algorithm 1.

After the topology of collaborative relay network is generated, we randomly schedule the transmissions and calculate the
total time saved $\delta$ by the concurrent transmissions. Per-client throughput and total throughput in CRN can be computed. The simulation results are discussed in Section VI.

V. DISTRIBUTED COLLABORATIVE RELAY PROTOCOL

In this section, we propose a distributed collaborative relay protocol for CRN to adapt to the real network environment.

A. The Arrival of A New Client

When a new client $v_i$ comes into the current CRN, it firstly tries to establish a direct communication link with AP. Then, we can get $T(v_i)$ easily, which is equal to $t(AP, v_i)$.

In order to improve its own and the whole network’s throughput, the client wants to get a better $T(v_i)$. It broadcasts a “join_request” message including $T(v_i)$. Any client $v_k$ who receives the “join_request” message from $v_i$ can get $t(v_i, v_k)$ by analyzing signal quality. It compares $T(v_i)$ with the sum of its current $T(v_k)$ and $t(v_i, v_k)$. If $T(v_k) + t(v_i, v_k) < T(v_i)$, there is throughput gain for $v_k$ to relay traffic for $v_i$. Then, a “join_reply” message including $T(v_k) + t(v_i, v_k)$ is sent back by $v_k$.

After a specific time window, $v_i$ can get all the “join_reply” messages with the sum of $T(v_k)$ and $t(v_i, v_k)$ smaller than $T(v_i)$. It then chooses the best one as the new $T(v_i)$. A “join_confirm” message is sent to the relay client $v_k^*$, which is selected by $v_i$. All the uplink traffic of $v_i$ is also sent to client $v_k^*$.

Then, $v_k^*$ starts to relay traffic for $v_i$ and resends a “join_confirm” message to its parent node, which can be AP or another relay client that relays traffic for client $v_k^*$. The downlink relay route is established once the “join_confirm” message is delivered to AP.

When the new communication link is established, $v_i$ broadcasts an “info_update” message including $T(v_i)$. Any client who want $v_i$ to relay traffic can also send “join_confirm” message to it.

B. The Leaving of A Current Client

When a current client $v_i$ wants to leave, it firstly sends a “leave_request” message to AP. If it is a relay node, it also sends a “leave_request” message to its children clients. After a certain period of time, $v_i$ can leave the network.

Once AP receives the “leave_request” message from $v_i$, it stops sending packets to it. All the packets sent to other clients through $v_i$ are sent by the direct communication link.

Similarly, once a client $v_k$ receives “leave_request” message from its relay node, it stops transmitting data to the relay node immediately. All of the data is sent via the link to AP directly.

After a random period of time, the clients who switch from relay route to the direct link broadcast “join_request” messages, and then follow the protocol for the new clients to establish a new relay route.

VI. SIMULATION RESULTS

In order to evaluate the performance of CRN, Two simulators using C++ programming language have been developed. The first one uses the centralized algorithm proposed in section IV. The other one uses the distributed protocol proposed in section V. 802.11g protocol is assumed to be used in the network, and all the nodes transmit data at the power of 20dbm. The packet length $C$ is set to 1512 bytes, and overhead for each packet is assumed to be 94\mu s.

In the first simulation, we use the centralized algorithm proposed in section IV, which is the foundation of our distributed protocol. 5-40 clients are randomly placed to the service range (64 meters) of the AP. As is shown in Fig.3, when the number of nodes is low, the throughput improvement for both 1-hop relay and CRN is limited, because there are less chance for the clients to find out a suitable candidate for relaying traffic. When the number of nodes rises, CRN shows its advantage, the throughput can be up to 42% higher than original WLAN systems on average.

In order to study the effort of multi-hop relay, we make a comparison between different selections by clients. From Fig.4, comparing CRN (M) with 1-hop (O) network, we could see more clients in CRN using relay nodes to increase the throughput. When the total number of clients grows, there are more chances for the clients to get a better multi-hop relay chain instead of direct link or 1-hop relay.

By using multi-hop relay, there are more opportunities for the links to transmit concurrently. As is shown in Fig.5, with the number of clients increasing, up to 39% percent of the communication links can be activated with at least another link at the same time. If we could take full advantage of concurrent transmissions, the performance could be further improved.

For the distributed protocol, whether it can use minimum control packets to respond to the network change and achieve the best topology is the point we want to verify. We generate an event list which contains 100 events. 65 of them are the events that new client arrives, and 35 of them are the events that client leaves.

In the simulation, all of these events happen one by one, and the interval time between the two consequent events is one second. As the simulation results shown in Fig.6, we observe...
that the network can handle the network change and improve the throughput a lot.

Simulation results show that totally 284 control messages are sent to handle these 100 network events. Compared to the high network data transmission rate, the control messages are sent at a relatively low level, and can be accepted by the current WLAN network. What’s more, the distributed protocol always achieves the same solution as the centralized algorithm we proposed for every event in the simulation, if we assume that every client responds to the “info_update” message when it finds throughput gain.

VII. CONCLUSION

In this paper, we present a new approach, which uses multi-hop client collaborative relays, and allows concurrent transmissions, to alleviate the rate anomaly problem in IEEE 802.11 WLANs. A centralized algorithm is proposed and proved to be optimal when concurrent transmissions are not optimized. Based on this centralized algorithm, we develop a distributed protocol to establish and maintain a collaborative relay network. The simulation results show both our algorithm and protocol can achieve substantial throughput improvement.

As a logical next step to this work, we are going to implement our proposed protocol in a prototype system. We are also exploring the methods to decrease the packet overhead to further improve the performance of CRN.

ACKNOWLEDGMENT

This work was supported by grants from NSF China [Grant No. 60633020 and 60970117].

REFERENCES